

WHAT IS CLAIMED IS:

- 1 1. A processor-implemented method for analyzing operations of an emulated
2 input-output processor (IOP), comprising:
3 emulating instructions native to a first type of instruction processor on a
4 second-type instruction processor;
5 executing by an instruction processor emulator an operating system
6 including instructions native to the first type of instruction processor, and
7 including instructions that write input/output (IO) requests to a memory
8 arrangement in response to IO functions invoked by a program;
9 emulating IOP processing of IO requests with an IOP emulator executable
10 on the second-type processor, including processing IO requests from the memory
11 arrangement and maintaining in the memory arrangement a first set of data
12 structures used in processing the IO requests;
13 storing state data currently contained in the data structures on a retentive
14 storage device; and
15 reading state data from the retentive storage responsive to a user input
16 control and displaying the data.
- 1 2. The method of claim 1, further comprising:
2 writing the state data read from retentive storage in a second set of data
3 structures that are of the same type as the first set of data structures
4 reading selected data from the second set of structures responsive to user-
5 input controls and displaying the selected data.
- 1 3. The method of claim 1, wherein the selected data include data that
2 describe each IO request.

- 1 4. The method of claim 1, further comprising:
2 maintaining a cache of data from a storage device by the IOP emulator;
3 and
4 accumulating statistics describing management of the cache, wherein the
5 selected data include the statistics describing management of the cache.
- 1 5. The method of claim 4, wherein the selected data include respective
2 numbers of read requests and write requests.
- 1 6. The method of claim 5, wherein the selected data include respective
2 numbers of cache hits and cache misses.
- 1 7. The method of claim 6, wherein the selected data include respective
2 numbers of read requests of different sizes and respective numbers of write
3 requests of different sizes.
- 1 8. The method of claim 1, wherein the data structures include a first queue of
2 in-process IO requests and a second queue of completed IO requests, and the
3 selected data include data from the first and second queues.
- 1 9. The method of claim 1, further comprising:
2 emulating IOP processing of IO requests with a plurality of IOP emulators
3 executable on the second-type processor, and each IOP emulator maintaining in
4 the memory arrangement respective data structures used in processing IO
5 requests directed to that IOP emulator;
6 wherein the selected data include data from the data structures associated
7 with an IOP emulator in response to a user-selected IOP emulator.
- 1 10. An apparatus for analyzing operations of an emulated input-output
2 processor, comprising:

3 means for emulating instructions native to a first type of instruction
4 processor on a second-type instruction processor, the instructions including
5 operating system instructions that write input/output (IO) requests to a memory
6 arrangement in response to IO functions invoked by a program;

7 means for emulating on the second-type processor, IOP processing of IO
8 requests from the memory arrangement and maintaining in the memory
9 arrangement a first set of data structures used in processing the IO requests;

10 means for storing state data currently contained in the data structures on a
11 retentive storage device; and

12 means responsive to a user input control for reading state data from the
13 retentive storage and displaying the state data.

1 11. The apparatus of claim 10, further comprising:

2 means for maintaining a cache of data from a storage device by the means
3 for emulating IOP processing of IO requests; and

4 means for accumulating statistics describing management of the cache,
5 wherein the selected data include the statistics describing management of the
6 cache.

1 12. A processor-implemented method for analyzing operations of an emulated
2 input-output processor, comprising:

3 emulating instructions native to a first type of instruction processor on a
4 second-type instruction processor;

5 executing by an instruction processor emulator, an operating system
6 including instructions native to the first type of instruction processor, the
7 operating system including instructions that write input/output (IO) requests to
8 a memory arrangement and read input data from the memory arrangement in
9 response to IO functions invoked by a program;

10 emulating IOP processing of IO requests with an IOP emulator executable
11 on the second-type processor, including processing IO requests from the memory

12 arrangement and maintaining in the memory arrangement a first set of data
13 structures used in processing the IO requests;
14 displaying, via a first tool, selected data from the first set of data structures
15 in response to a user input control;
16 storing state data currently contained in the data structures on a retentive
17 storage device;
18 reading state data from the retentive storage responsive to a user input
19 control and writing the state data to a second set of data structures in the
20 memory arrangement, wherein the second set of data structures are identical in
21 structure to the first set of data structures; and
22 displaying, via the first tool, selected data from the second set of data
23 structures in response to a user input control.

1 13. The method of claim 12, wherein the selected data from the first and
2 second sets of data structures include data that describe each IO request.

1 14. The method of claim 12, further comprising:
2 maintaining a cache of data from a storage device by the IOP emulator;
3 and
4 accumulating in the first set of data structures statistics that describe
5 management of the cache, wherein the selected data from the first and second
6 sets of data structures include the statistics describing management of the cache.

1 15. The method of claim 14, wherein the selected data from the first and
2 second sets of data structures include respective numbers of read requests and
3 write requests.

1 16. The method of claim 15, wherein the selected data include respective
2 numbers of cache hits and cache misses.

1 17. The method of claim 16, wherein the selected data from the first and
2 second sets of data structures include respective numbers of read requests of
3 different sizes and respective numbers of write requests of different sizes.

1 18. The method of claim 12, wherein the first and second sets of data
2 structures include a first queue of in-process IO requests and a second queue of
3 completed IO requests, and the selected data from the first and second sets of
4 data structures include data from the first and second queues.

1 19. The method of claim 12, further comprising:
2 emulating IOP processing of IO requests with a plurality of IOP emulators
3 executable on the second-type processor, and each IOP emulator maintaining in
4 the memory arrangement respective data structures used in processing IO
5 requests directed to that IOP emulator;
6 wherein the selected data from the first and second sets of data structures
7 include data from the data structures associated with an IOP emulator in
8 response to a user-selected IOP emulator.